APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE

e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 879-896
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

UTHM Losthub – Mobile Application for Lost and Found Items in UTHM

Abdul Razlan Khairy Abdul Rahim[1], Siti Hajar Arbain[1*]

[1] *Faculty of Computer Science and Information Technology,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: sitihajara@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.047

Article Info

Received: 15 June 2025
Accepted: 22 June 2026
Available online: 30 June 2026

Keywords
UTHM Losthub, Lost and found, Android development

Abstract

This paper presents the design and development of *UTHM LostHub*, a mobile application aimed at addressing the inefficiencies in managing lost and found items at Universiti Tun Hussein Onn Malaysia (UTHM). The existing methods such as noticeboards and social media were found to be unstructured and unreliable. UTHM LostHub was developed as a centralized, user-friendly platform using Android Studio and Firebase, enabling students and staff to report, search, and manage lost and found items efficiently. The project adopted a prototyping model, allowing iterative development and integration of user feedback. Key features include secure authentication via institutional email, advanced item filtering, and direct communication between users. Comparative analysis with similar systems demonstrated the advantages of LostHub in terms of usability and functionality. The application underwent extensive functional and user acceptance testing (UAT), with results showing high user satisfaction and positive feedback on its interface and core features. While the system performs well, minor limitations were identified, including the absence of a password recovery feature and occasional performance issues. Future improvements will focus on optimizing system performance and enhancing accessibility. The UTHM LostHub app stands as a practical digital solution for improving campus-wide lost and found item management.

1. Introduction

In today's fast-paced academic environment, managing lost and found items efficiently is essential for maintaining a productive and secure campus experience, particularly at institutions such as Universiti Tun Hussein Onn Malaysia (UTHM). Traditional methods of handling lost items such as physical notice boards or scattered social media posts are often unstructured, unreliable, and time-consuming. These informal systems make it difficult for students and staff to report, track, and recover missing belongings. Studies show that over 50% of university students lose valuable items such as student ID cards, electronic devices, and wallets annually, resulting in stress and academic disruption [1]. Without a centralized solution, UTHM faces ongoing challenges in managing lost property and supporting its community in recovering their belongings promptly and securely.

This is an open access article under the CC BY-NC-SA 4.0 license.

The UTHM LostHub project addresses these concerns by introducing a dedicated mobile application designed to streamline the management of lost and found items on campus. The system enables students and staff to easily report lost or found items, browse existing reports, filter listings by category, and communicate directly with others to coordinate item returns. By leveraging mobile technology, LostHub removes the limitations of time and location, providing users with a reliable and efficient digital platform to handle lost and found cases anytime, anywhere. The application also ensures that all user interactions are secure and accessible only to authorized individuals via institutional email login, further enhancing data protection and system integrity [2].

The first objective of this project is to design UTHM LostHub using an object-oriented approach. This involves establishing a structured foundation, including the application's architecture, interface, and system logic, based on object-oriented principles. By breaking down complex features into modular and reusable components, this design approach improves system scalability, maintainability, and development efficiency. The second objective is to develop a campus-focused platform using a mobile app-based solution tailored to the needs of UTHM students and staff. The system will promote seamless communication and user engagement around lost and found matters, enhancing its value as a reliable and user-friendly utility for the campus community. Lastly, the third objective is to test the functionality of the developed system through User Acceptance Testing (UAT). This testing phase will involve UTHM students and staff as actual users to evaluate the usability and performance of the app, gather feedback, and ensure that the application aligns with real-world expectations.

The main issue at UTHM is the absence of an organized, institution-wide platform for lost and found item management. Current approaches lack consistency and traceability, and they provide no easy way for users to follow up on reported items or confirm ownership. LostHub solves this by offering a centralized, reducing manual intervention and enabling quicker item recovery. It also minimizes staff workload and promotes community responsibility through direct user engagement. Additionally, the mobile-first approach aligns with current trends in Android application development, ensuring that the system benefits from modern design principles and evolving mobile security frameworks [3].

In this documentation, Section 2 discusses related works and existing systems used in similar environments, providing a comparative analysis with the UTHM LostHub solution. Section 3 presents the methodology applied to the system's development. Section 4 outlines result and discussion, including major features, implementation details, and testing outcomes that form the foundation of the UTHM LostHub mobile application.

2. Related Work

This section provides a literature review on topics related to android technology, software development life cycle, and existing comparative systems. The literature review aims to gain a comprehensive understanding and discuss the project background about the proposed system.

2.1 Overview of Android Technology

Android, as a leading mobile operating system, provides a robust platform for developing scalable and resource-efficient applications. Its open-source nature and comprehensive SDK facilitate cross-platform development and debugging. The flexibility of Android allows developers to create tailored solutions, while its integration capabilities enable seamless communication between devices. However, its popularity also introduces security vulnerabilities, necessitating robust measures to ensure data integrity and user trust. Studies suggest incorporating static and dynamic security measures during development to mitigate these risks and build a trusted application [4][5].

2.2 Overview of Software Development Life Cycle

The Software Development Life Cycle (SDLC) provides a structured approach to software development, covering requirements gathering, design, implementation, testing, and maintenance. Various models of SDLC, such as Agile and Waterfall, cater to different project needs. Agile, in particular, allows iterative development and feedback integration, making it ideal for user-focused applications like LostHub. A significant advantage of SDLC is its emphasis on systematic planning, which ensures clarity and efficiency throughout the development process [6].

Furthermore, integrating security at every stage is increasingly recognized as essential for creating reliable and robust software. Traditional SDLC approaches often introduced security considerations only at later stages, which can lead to costly and complex revisions. A secure SDLC (S-SDLC) model, as proposed by Mohino et al., incorporates security requirements early in the development process especially within Agile methodologies to reduce vulnerabilities and improve overall system resilience [7].

2.3 Mafqodat – A Secure and Safe Mobile Application to Retrieve Lost Items in King Saud University

Mafqodat, developed for King Saud University, addresses lost and found challenges by providing features such as secure item reporting, discovery locations, and profile management. Its Agile development methodology ensures iterative refinement, leading to a user-friendly platform. However, Mafqodat's basic search functionality and lack of direct communication tools limit its effectiveness. Future enhancements could include advanced filtering options and integration with messaging systems to improve user interactions and item recovery rates [8].

2.4 Lost Things – Mobile Application for Lost and Found Items in King Khalid University

Lost Things leverages the Flutter framework and Firebase to deliver a mobile solution for managing lost and found items at King Khalid University. The application prioritizes simplicity and user accessibility, offering basic item reporting and search functionalities. However, its limitations include a lack of personalized dashboards, advanced search filters, and direct communication features. These shortcomings make the recovery process less efficient, especially for users needing detailed item searches. Enhancing navigation and incorporating robust communication tools could significantly elevate its functionality and usability [9].

2.5 University of Baghdad – Smartphone Application for Managing Missed and Found Belongings

The University of Baghdad's application adopts a three-layer architecture, encompassing presentation, business logic, and data layers, to manage lost and found items. This design enables efficient data synchronization and complaint registration. However, similar to other systems, it lacks features like direct communication tools and secure login mechanisms. These omissions reduce its ability to provide a seamless user experience. By addressing these limitations, such as adding advanced filter options and integrating secure authentication methods, the application could better serve its users and streamline item recovery processes [10].

2.6 Comparison with the Existing Systems

Table 1 shows the comparison between the existing systems Mafqodat (King Saud University), Lost Things (King Khalid University), and the University of Baghdad App and the proposed UTHM LostHub System. This table compares the characteristics of each system, including user features, communication methods, filtering capabilities, and overall usability. The comparison approach is based on identifying key or "hot" features commonly used in mobile app evaluations [11].

Table 1 *Systems Comparison*

Features	Mafqodat (King Saud University)	Lost Things (King Khalid University)	University of Baghdad App	UTHM LostHub
Register, create an account and log in.	√	√	√	√
Unique Email Registration	√	X	X	√
Detailed Post Creation	√	√	√	√
Advanced Filter Options	X	X	X	√
Direct Contact Features	X	X	X	√
Personalized Dashboards	X	X	X	√
Discovery location	√	X	X	√

User-Friendly Design	Moderate	Poor	Moderate	Excellent

Legend: √ = Yes ； X= No

3. Methodology

The UTHM LostHub project adopted the prototyping model, a methodology well-suited for iterative and user-focused development. This approach emphasizes building early working versions of the application for immediate testing and refinement based on continuous user feedback. By incorporating user insights at every stage, it ensures the final product meets real needs and expectations. Prototypes are essential to creating useful products and fostering creative ideas [12]. As shown in Figure 1 Prototyping Model.

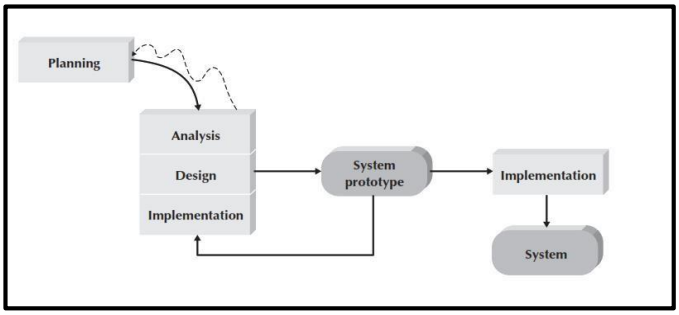


Fig. 1 *Prototyping Model*

3.1 Planning Phase

The project began with defining clear objectives, such as creating a centralized platform for managing lost and found items within the UTHM campus. Emphasis was placed on streamlining the recovery process through a structured and accessible mobile application. A detailed Gantt chart was prepared to outline the key milestones, ensuring that all development activities progressed according to schedule.

A critical decision during this phase was the adoption of the Prototyping Model. This model was selected for its ability to adapt to iterative feedback and allow frequent updates to system features based on real user input. The planning phase also included a thorough review of the UTHM environment, highlighting the specific needs and challenges faced by students and staff in managing lost and found items. These insights served as a guiding framework for the subsequent phases.

3.2 Analysis Phase

The analysis phase focused on identifying and addressing existing gaps in current lost and found practices at UTHM. A comparative evaluation of existing systems such as Mafqodat, Lost Things, and the University of Baghdad's Lost and Found app provided valuable insights into common limitations and potential design improvements. These systems highlighted the lack of real-time communication tools, inadequate filtering features, and the absence of centralized databases.

In parallel, user requirements were collected from UTHM students and staff to ensure that the system would cater directly to their needs. These were derived through observations, interviews, and feedback from potential users. The gathered requirements were categorized into user, functional, and non-functional requirements, which form the basis for system development. Table 2 presents the user requirements, detailing what the system should allow users to perform.

Table 2 *User Requirements*

User	Requirement

Student/Staff	• Enable to register an account using a verified UTHM email. • Enable to log in to the system. • Enable to report a lost item by entering item details and uploading images. • Enable to submit a found item by entering item details and uploading images. • Enable to initiate communication with item finders or owners through SMS or call options. • Enable to match for lost or found items using filters • Enable to delete reported lost or found items. • Enable to view item details from search results.

In addition, Table 3 outlines the functional requirements, specifying the main features and operations the UTHM LostHub application must support.

Table 3 *Functional Requirements*

Function	Requirement
Register and Log In	• The system shall allow users (UTHM students and staff) to register using their UTHM email address and credentials. • The system shall authenticate users to ensure only authorized users can access the platform. • The system shall verify the user's identity during login to provide secure access.
Report Lost Item	• The system shall allow users to report a lost item by entering details. • The system shall allow users to upload images for better identification of lost items.
Submit Found Item	• The system shall allow users to submit details for found items. • The system shall allow users to upload images for found items to assist with identification.
Filter Items	• The system shall allow users to search for lost and found items using filters • The system shall provide search filters to refine results for easier item discovery. • The system shall allow users to delete lost or found items that have been recovered or are no longer needed. • The system shall allow users to delete items directly after a search result or upon item recovery (Optional).
Communicate	• The system shall validate the contact information before enabling communication. • The system shall enable users to contact item finders or owners directly through integrated SMS or call options. • The system shall allow communication initiation after search results to facilitate the return of items (Optional).

A critical aspect of this phase involved the development of Unified Modeling Language (UML) diagrams. These diagrams were used to visualize system workflows and interactions. The use case diagram, as shown in Figure 2 below, depicted key system functionalities and user roles. Additional diagrams include activity diagrams to illustrate procedural flows and sequence diagrams to show the step-by-step logic of user actions. A class diagram was also developed to define the relationships between system entities and their data attributes. Additionally, a system architecture diagram was created to illustrate the overall structure of the application. Both the class diagram and system architecture diagram are provided in Appendix.

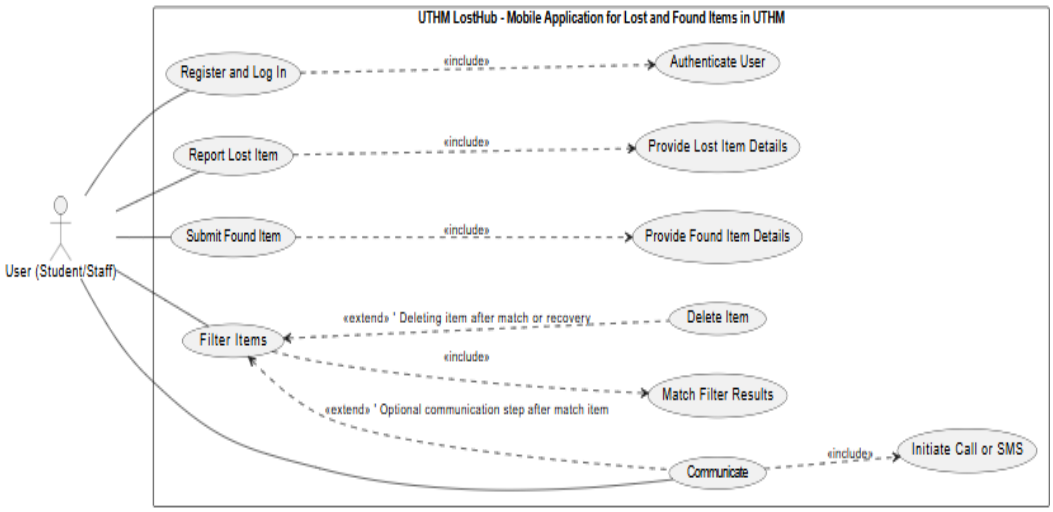


Fig. 2 *Use Case Diagram*

This phase also included the specification of hardware and software requirements to ensure the application's compatibility, performance, and scalability on Android devices. The analysis results provided a strong foundation for the following design phase, ensuring that the application met both user expectations and technical feasibility.

3.3 Design Phase

During the design phase, the requirements gathered were translated into a comprehensive development blueprint. Special attention was given to the application's interface and data structure to ensure both clarity and scalability. Wireframes were developed to visualize the layout and flow of user interfaces, focusing on simplicity and intuitive navigation. Key interfaces included registration, login, item submission, and filtering features. On the backend, a Firebase database schema was designed to manage user data, item records, and communication histories efficiently. The schema ensured that data operations such as submission, retrieval, and deletion were executed smoothly. The previously developed class diagram served as a structural foundation, showing the interrelation between data entities such as users, lost items, and found items.

To further document the system's internal logic, a data dictionary was prepared, detailing the attributes of each database field. The design choices emphasized maintainability, data integrity, and the ability to incorporate future enhancements.

3.4 Implementation Phase

The implementation phase involved the development of the mobile application using Java in Android Studio for the front-end and Firebase for the backend. This technology stack enabled real-time data synchronization, secure user authentication, and seamless integration of dynamic features. Firebase services also supported image storage for item reports and robust cloud data management.

Two prototypes were developed during this phase. The first prototype included core functionalities such as user registration and item filtering. It was tested internally to validate basic system operations. Based on feedback from this initial version, improvements were implemented, and additional features were integrated into the second prototype, developed in February 2025. This version introduced enhanced filtering, communication tools (via SMS and call integration), and more refined user interface elements. This iterative development process reflects the strengths of the Prototyping Model, which emphasizes user involvement and continuous refinement [13].

Throughout the implementation phase, user feedback played a pivotal role. Internal and external user acceptance testing (UAT) was conducted to evaluate the system's functionality, usability, and performance. The feedback obtained during UAT sessions was instrumental in identifying issues and refining the application to meet user expectations. Table 4 shows the software development activities and their tasks for UTHM LostHub.

Phase	Task	Output

Table 4 *Software Development Activities and Their Task*

Planning	Study on project background • Define • Problem Statement • Objective • Scope • Expected results of the project • Project Significance Decide on development model: Prototyping model	• Project Proposal • Gantt Chart • List of Project Objectives • Project Scope Definition • Expected Results Summary • Project Significance Explanation • Selected Development Model
Analysis	Study on the existing system • King Saud University – Mafqodat: A Secure and Safe Mobile Application to Retrieve Lost Items • Lost Things – Mobile Application for Lost and Found Items in King Khalid University • King Saud University of Baghdad – Smartphone Application for Managing Missed and Found Belogings System Analysis	• Problem Statement • Comparison table between existing system and proposed system • User Requirement • Functional Requirement • Non- Functional Requirement • UML Diagram
Design	• System Design • Craft detail wireframes	• Wireframes • Database Schema • Interface Design

Implementation	• Coding on Android Studio
	• Connect with firebase
	• Alpha and Beta test the system
		• Source Code (Java/XML files for Android)
		• Firebase-integrated mobile app
		• Functional modules implemented:
		  • Register and Log in
		  • Report Lost Item
		  • Submit Found Item
		  • Filter Items
		  • Communicate
		• Test Case Document
		• System Testing Result
		• User Acceptance Testing (UAT)

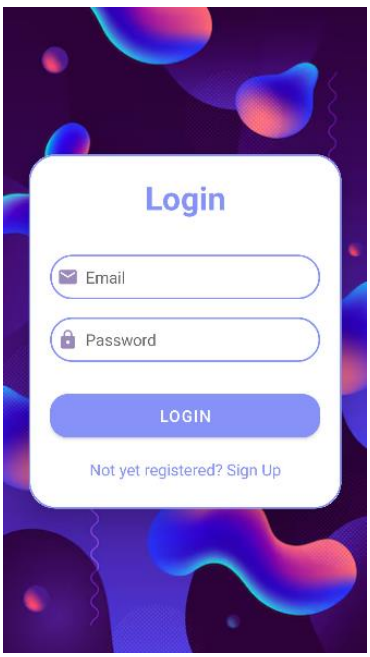
4. Result and Discussion

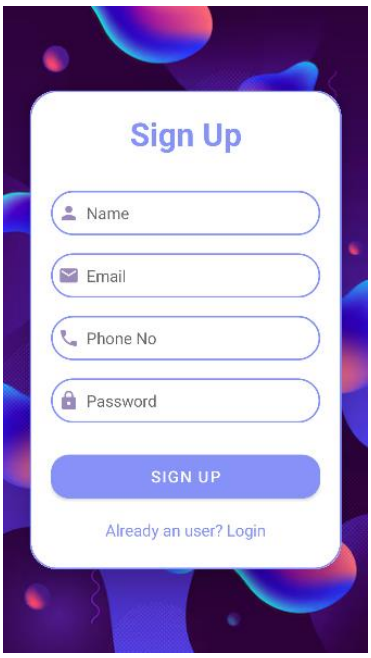
This section presents the implementation of UTHM LostHub in Android Studio using Java as the primary programming language, along with the system's user interface design. The functional testing results and User Acceptance Testing (UAT) outcomes are analyzed to evaluate the application's performance, usability, and effectiveness in addressing lost-and-found scenarios within the UTHM community.

4.1 Implementation

The UTHM LostHub application is implemented using integrated libraries and frameworks, with the primary objective of translating the defined specifications and requirements into functional code. The application was developed utilizing the Android Studio Integrated Development Environment (IDE). User interface designs were crafted using XML, while the backend logic was written in the Java programming language. Firebase was integrated to manage the database, ensuring real-time data synchronization, authentication, and secure storage.

Figure 3 (a) illustrates the login user interface. This interface allows users to log in to their accounts by providing their UTHM email and password. Upon successful validation of these credentials against Firebase authentication, the user gains access to the application. Figure 3 (b) displays the registration form user interface. New users can create an account by providing their full name, UTHM email address, phone number, and a password. Once all required information is submitted and validated, the registration is successful, and the user is directed back to the login page to access their newly created account.

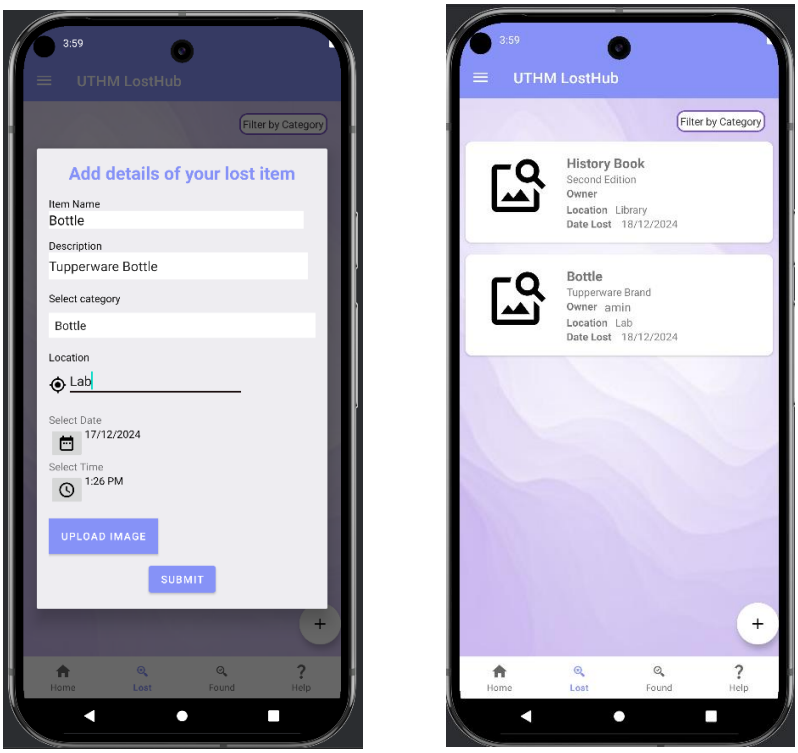




(a) **(b)**

Fig. 3 *Register and Log in Module (a) Login User Interface; (b) Register User Interface*

For the Report Lost Item module, Figure 4 (a) shows the user interface to allow users to input all the relevant data for a lost item such as item name, category, description, location, date, time, and an optional image. Once the user submits these details, the data is processed and saved to the Firebase database. Figure 4 (b) illustrates how lost items are displayed in the Lost Interface after successful submission. The code snippet for retrieving and displaying lost item details, as shown in Figure 5 demonstrates how an itemRef can be used to fetch the document and display lost details.



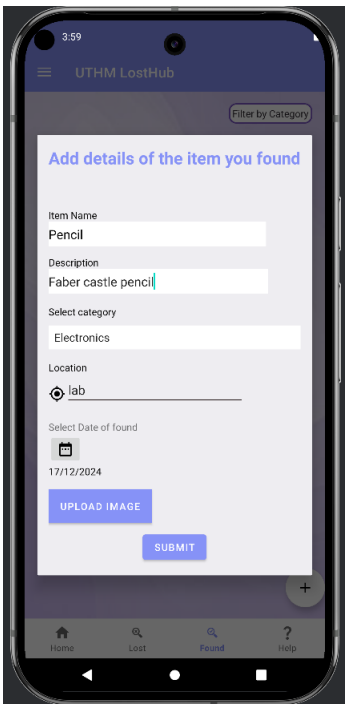
(a) **(b)**

Fig. 4 *Report Lost Item Module (a) Lost Form Interface; (b) Lost Interface*

```
itemRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
        if (documentSnapshot.exists()) {
            String imageUrl = documentSnapshot.getString( field: "imageURI");
            String itemTitle = documentSnapshot.getString( field: "itemName");
            String itemAddress = documentSnapshot.getString( field: "location");
            String itemMail = documentSnapshot.getString( field: "email");
            String itemDescription = documentSnapshot.getString( field: "description");
            String itemOwnerName = documentSnapshot.getString( field: "ownerName");
            String itemDateLost = documentSnapshot.getString( field: "dateLost");
            String phone = String.valueOf(documentSnapshot.getLong( field: "phnum")) ;
            String itemtimeLost = documentSnapshot.getString( field: "timeLost");
            String categ = documentSnapshot.getString( field: "category");
            // Load the image using Glide and adjust the ImageView size
            if (imageUrl != null && !imageUrl.isEmpty()) {
                Glide.with( activity: LostDetails.this) RequestManager
                        .load(imageUrl) RequestBuilder<Drawable>
                        .placeholder(R.drawable.placeholder_image)
                        .error(R.drawable.baseline_image_search_24)
                        .into(img);
            }
            // Set data to TextViews
            title.setText(itemTitle);
            address.setText(itemAddress);
            mail.setText(itemMail);
            description.setText(itemDescription);
            ownerName.setText(itemOwnerName);
```

Fig. 5 *Fetch The Document and Display Lost Details*

Similarly, for the Submit Found Item module, Figure 6 (a) presents the user interface where individuals can report items they have found. This interface enables users to provide comprehensive details about the found item, including its name, category, description, location of discovery, date, time, contact information, and an optional image for better identification. Once these details are submitted, the system processes and stores the data securely in the Firebase database. Figure 6 (b) demonstrates how these reported found items are subsequently displayed in the Found Interface, allowing other users to browse and potentially identify their lost belongings. The code snippet for retrieving and displaying found item details, also shown in Figure 7 illustrates the use of an itemRef to fetch the document and display found details.



(a)

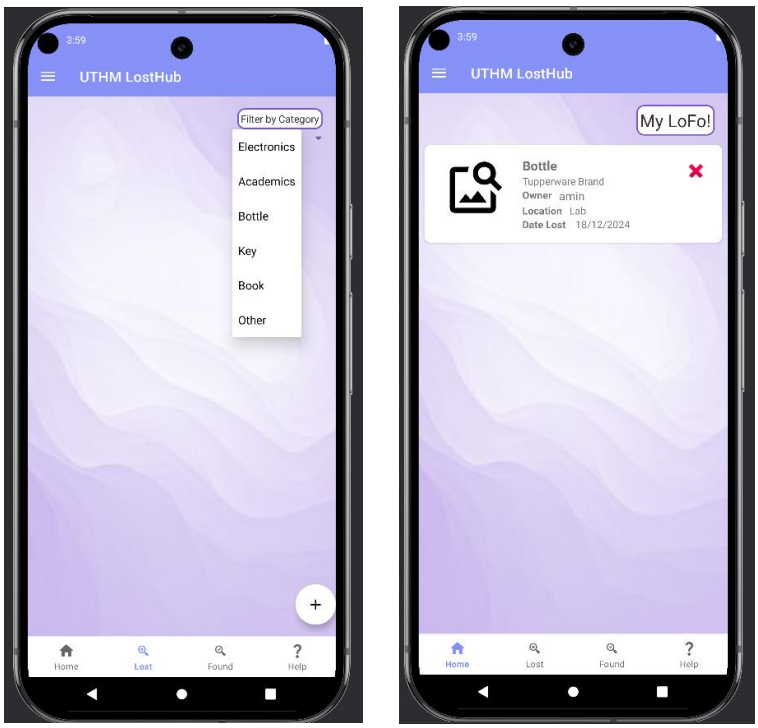
(b)

Fig. 6 *Submit Found Item Module (a) Found Form Interface; (b) Found Interface*

```
itemRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
        if (documentSnapshot.exists()) {
            String imageUrl = documentSnapshot.getString( field: "imageURI");
            String itemName = documentSnapshot.getString( field: "itemName");
            String itemLocation = documentSnapshot.getString( field: "location");
            String itemEmail = documentSnapshot.getString( field: "email");
            String itemDescription = documentSnapshot.getString( field: "description");
            String itemOwner = documentSnapshot.getString( field: "finderName");
            String itemPhnum = documentSnapshot.getString( field: "phnum");
            String itemDate = documentSnapshot.getString( field: "dateFound");
            String itemCategory = documentSnapshot.getString( field: "category");
            // Load the image using Glide and adjust the ImageView size
            if (imageUrl != null && !imageUrl.isEmpty()) {
                Glide.with( activity: FoundDetails.this) RequestManager
                        .load(imageUrl) RequestBuilder<Drawable>
                        .placeholder(R.drawable.placeholder_image)
                        .error(R.drawable.baseline_image_search_24)
                        .into(img);
            }
            // Set the text for the attributes in the .xml file
            title.setText(itemName);
            address.setText(itemLocation);
            dateFound.setText(itemDate);
            description.setText(itemDescription);
            email.setText(itemEmail);
            finderName.setText(itemOwner);
```

Fig. 7 *Fetch The Document and Display Found Details*

To enhance user experience and facilitate item management, the application integrates both the Filter Items module and the My Item section. The Filter Items module, depicted in Figure 8 (a) provides users with an interface to efficiently search through the vast repository of lost and found items. Users can filter items by item category. When a user applies a filter, the system dynamically retrieves and displays only the items that match the specified criteria, ensuring a streamlined and relevant browsing experience. Complementing this, the My Item section, showcased in Figure 8 (b) offers a dedicated area for users to manage the items they have personally reported, whether lost or found. This interface lists all items posted by the current user and includes a crucial feature: the ability to delete an item. Users can remove a previously reported item from the database, for instance, if a lost item has been recovered or a found item has been claimed, thereby ensuring data accuracy and empowering users to maintain control over their posted information within the application.



(a) **(b)**

Fig. 8 *Filter Items Module (a) Filter Items Interface; (b) My Item Interface*

Finally, the Communicate Module enables users to connect with item reporters directly from the application. Its functionality is driven by specific code snippets. As shown in Figure 9 (a) Handle Call and SMS for Lost Item and Figure 9 (b) Handle Call and SMS for Found Item, a code snippet demonstrates how an onClickListener is set for both the Call button and the SMS button. This code allows the application to initiate a phone call or send an SMS message to the item reporter's contact number. This function provides a direct and efficient way for users to communicate regarding lost or found items, facilitating the return process.

```
call.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // When the Call button is clicked, open the phone dialer
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse( uriString: "tel:" + phoneNum));
        startActivity(intent);
    }
});

sms.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {

        // When the SMS button is clicked, open the SMS app
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse( uriString: "sms:" + phoneNum));
        intent.putExtra( name: "sms_body",  value: "Hello, I want to inquire about your lost item.");
        startActivity(intent);
    }
});
```

(a)

```
callBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // When the Call button is clicked, open the phone dialer
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse( uriString: "tel:" + phnum));
        startActivity(intent);
    }
});

// Set an onClickListener for the SMS button
smsBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // When the SMS button is clicked, open the SMS app
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse( uriString: "sms:" + phnum));
        intent.putExtra( name: "sms_body",  value: "Hello, I want to inquire about item you found.");
        startActivity(intent);
    }
});
```

(b)

Fig. 9 *Communication Module (a) Handle Call and SMS for Lost Item;*

(b) Handle Call and SMS for Found Item

4.2 Functional Testing

Functional testing plays a critical role in verifying that the UTHM LostHub application meets the expected behavior outlined in its functional requirements. The testing focused on five core modules: Register and Login, Report Lost Item, Submit Found Item, Filter Items, and Communicate. Each module was tested using detailed test cases to validate both typical and edge-case scenarios involving different user roles within the system.

These test cases ensured proper system responses, accurate data validation, image upload capabilities, and communication functionality via SMS and calls. The results of the tests are shown in Table 5, confirming that each module performed successfully under all tested conditions.

Table 5 *Test Plan for Application Functionalities*

Function	Functionality	User	Actual Output
Register and Log in	Allows users to register using UTHM email, log in with credentials, and log out.	Student, Staff	Pass

Report Lost Item	Enables users to report lost items by entering item details and uploading images.	Student, Staff	Pass
Submit Found Item	Allows users to submit reports for found items with optional image uploads.	Student, Staff	Pass
Filter Items	Allows users to filter lost or found items by category to simplify the search process and delete their own reports if recovered.	Student, Staff	Pass
Communicate	Allows users to contact other users via SMS or call after item match.	Student, Staff	Pass

4.3 User Acceptance Testing

User Acceptance Testing (UAT) constitutes a critical phase within the software development lifecycle, specifically designed to affirm that a developed system aligns precisely with its stipulated functional requirements and the expectations of its end-users within practical, real-world contexts. For the UTHM LostHub mobile application, this comprehensive UAT process encompassed both alpha and beta testing phases, meticulously structured in adherence to the methodologies detailed in comparable development reports. Initially, alpha testing was executed internally by the development team, focusing on the identification and resolution of preliminary issues pertaining to system functionality and performance. Subsequently, the beta testing phase actively engaged a cohort of 30 end-users drawn from the Universiti Tun Hussein Onn Malaysia (UTHM) community, a demographic representative of the application's intended audience. These participants rigorously interacted with all core modules of the application, including Register and Log In, Report Lost Item, Submit Found Item, Filter Items, and Communicate, with their valuable feedback systematically collected via a structured User Acceptance Testing Form.

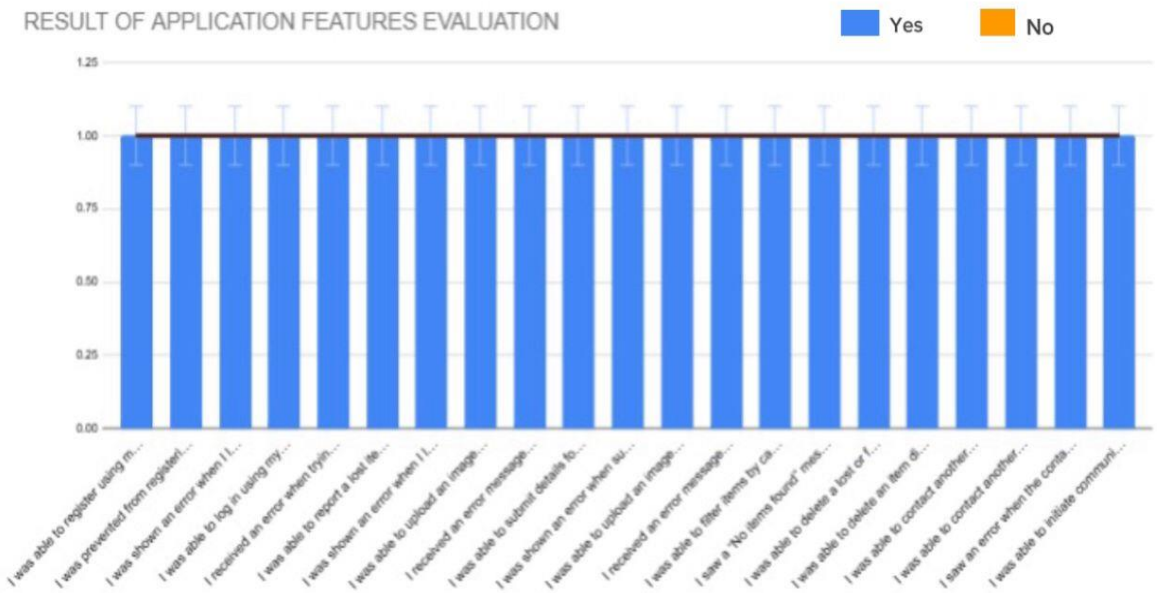


Fig. 10 *Result of Application Features Evaluation*

The cumulative evaluation of the application's features across all primary modules conclusively demonstrated a superior level of user satisfaction, thereby affirming that the UTHM LostHub application successfully met its intended functionalities. As depicted in Figure 10 Result of Application Feature Evaluation, each module exhibited robust performance, with all 30 participants (100%) consistently reporting seamless registration and login experiences using valid UTHM email credentials, complemented by accurate error handling for invalid

or duplicate inputs. Similarly, users universally succeeded (100%) in reporting lost items and submitting found items, providing all requisite details and uploading images without impediment, while also receiving appropriate error messages for missing fields or incompatible image formats. The filtering capabilities of the system also garnered unanimous positive feedback, enabling all participants to effectively sort items by category and accurately receiving "No items found" messages. Furthermore, the functionality allowing users to delete previously posted items, whether directly or subsequent to filtering or matching results, operated precisely as expected for every user.

Lastly, the communication module proved exceptionally effective, facilitating 100% successful initiation of contact with other users via integrated SMS and phone call features, with the system consistently validating contact information and providing communication options immediately following item filter matches. Collectively, these module-specific evaluations unequivocally confirmed that the UTHM LostHub application's features are resilient, fully functional, and fulfill their specified requirements with remarkable success across all user interactions.

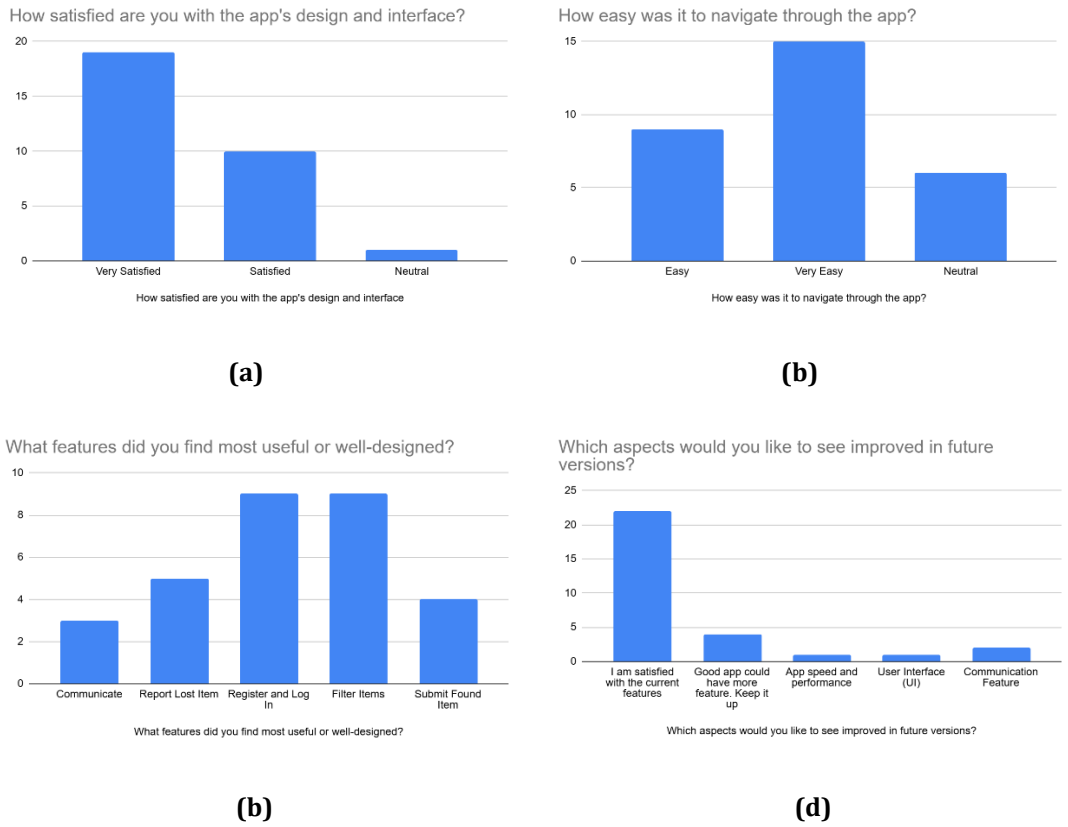


Fig. 11 *General Feedback (a) User satisfaction; (b) Navigation;*

(c) Most Useful Feature; (d) Improvement Suggestion

Beyond the granular assessment of individual modules, the general feedback furnished by participants yielded profound insights into the holistic user experience and illuminated prospective avenues for enhancement. As comprehensively illustrated in Figure 11 General Feedback, the overarching sentiment among users was markedly positive, with distinct strengths and areas for future refinement identified. User satisfaction with the app's design and interface was high as shown in Fig. 11 (a) User satisfaction, with 63.3% being "Very Satisfied" and 33.3% being "Satisfied," while only 3.3% found it "Neutral." A considerable majority of users perceived the application as intuitively navigable as presented in Fig. 11 (b) Navigation, with 46.7% categorizing it as "Very Easy" and 30% as "Easy," while a minor 23.3% found it "Neutral," underscoring a well-conceived and user-friendly structural flow. Participants concurrently highlighted "Register and Log In" (30%) and "Filter Items" (30%) as equally indispensable as seen in Fig. 11 (c) Most Useful Feature, thereby emphasizing the paramount importance of both secure accessibility and efficient information retrieval.

Additionally, "Report Lost Item" was acknowledged by 16.7% of users, "Submit Found Item" by 10%, and "Communicate" by another 10%, reflecting appreciation for these foundational functionalities. While a high overall satisfaction rate was observed (with 73.3% of users expressing contentment with the current features), other users also provided suggestions for future iterations. Key recommendations as detailed in Fig.

11 (d) Improvement Suggestion, included "App speed and performance improvement" (noted by 6.7% of respondents) and the desire for the incorporation of "more features" (also noted by 6.7% of respondents), indicating a demand for optimization and enhanced functionality to enrich the user experience. Minor comments also touched upon the "User Interface (UI)" (3.3%) and general encouragement to "Keep it up" (10%).

In conclusion, the UTHM LostHub application garnered highly encouraging feedback during its UAT phase, decisively achieving its foundational objectives and establishing itself as a practical and accessible tool for its target users. Consequently, future developmental efforts should strategically prioritize the optimization of overall system performance and the judicious integration of a 'forget password' feature as suggested by the invaluable user feedback.

5. Conclusion

This paper provides an overview of the development of the UTHM LostHub mobile application, designed to streamline the process of managing lost and found items within Universiti Tun Hussein Onn Malaysia. It begins by outlining the project objectives and challenges associated with the existing informal methods. A comparison with three similar systems from other universities highlights the advantages of LostHub's features. The development process followed a prototyping model, with each phase planning, analysis, design, and implementation presented systematically. The paper also discusses implementation modules, functional testing, and user acceptance testing (UAT), which confirmed the application's usability and effectiveness. The application offers a centralized platform for students and staff to report, search, and manage lost or found items efficiently, with integrated filtering and communication features. Test results showed strong user satisfaction, emphasizing the simplicity, clarity, and practicality of the interface.

Despite its strengths, the UTHM LostHub application has certain limitations. One of the main issues identified is the system's performance, especially during data-heavy operations such as filtering and retrieving item lists, which may result in minor delays for users. The lack of a "forgot password" feature also affects accessibility, particularly for users who misplace their login credentials.

To further enhance functionality and user experience, several improvements have been proposed. Implementing a password recovery option would significantly improve user accessibility. Optimizing the app's performance, especially for real-time data operations, would ensure smoother navigation and faster load times. These enhancements would not only strengthen the application's capabilities but also contribute to a more seamless and efficient campus-wide lost and found experience.

Acknowledgement

I would like to express my sincere appreciation to the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for their precious support during the research and development of this project. The guidance and resources provided by the faculty played a key role in the outcome of my work. My sincere thanks to the faculty for their strong commitment to academic excellence and research, which contributed significantly to the successful completion of this project.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Abdul Razlan Khairy Bin Abdul Rahim, Siti Hajar Binti Arbain;* ***data collection:*** *Siti Hajar Binti Arbain;* ***analysis and interpretation of results:*** *Abdul Razlan Khairy Bin Abdul Rahim, Siti Hajar Binti Arbain;* ***draft manuscript preparation****:Siti Hajar Binti Arbain. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] NotLost Press. (2020). 4.5 million items lost: The hidden cost of lost property at university. Retrieved June 30, 2023, from https://notlost.com/the-hidden cost-of lost-property-at university.

[2] Zborovsky, G., & Ambarova, P. (2018). Higher Education as a Factor of the Cities' Preservation in the Ural Macro-Region. Economy of Region.

[3] A. Sarkar, A. Goyal, D. Hicks, D. Sarkar and S. Hazra, "Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 73-79, doi: 10.1109/I SMAC47947.2019.9032440..
[4] Ahmed, O., & Sallow, A. (2017). Android Security: A Review. Academic Journal of Nawroz University, 6(3), 135–140. https://doi.org/10.25007/ajnu.v6n3a97.
[5] Bhat, P., & Dutta, K. (2019). A survey on various threats and current state of security in android platform. ACM Computing https://doi.org/10.1145/3301285.
[6] Iqbal, S. Z., platform. ACM Computing & Access, 6, 63859–63867. Idrees, M. (2017). SDLCModel:ANewModelForSowareDevelopmentLifeCycle(SDLC).
[7] Mohino, J. de V., Higuera, J. B., Higuera, J. R. B., & Montalvo, J. A. S. (2019). The application of a new secure software development life cycle (S-SDLC) with agile methodologies. Electronics (Switzerland), 8(11). (2019, January 9). https://doi.org/10.3390/electronics8111218.
[8] Allheeib, N. I. (2024, August). Mafqodat: A Secure and Safe Mobile Application to Retrieve Lost Items in Educational Institutes. In 2024 12th International Conference on Information and Communication Technology (ICoICT) (pp. 368-373). IEEE.
[9] Albishri, L. A., Alhefzi, L. A., Ziad, M. A. Y., & Almarie, L. S. A. (2023). Mobile Application for Lost and Found Items in King Khalid University. College of Computer Science, King Khalid University. Supervised by Dr. Quadri Noorulhasan Naveed.
[10] Salman, Z. A. J., & Athab, O. A. (2022). Smartphone application for managing missed and found belongings. MEST Journal, 10(1), 66-71.
[11] Malik, H., Shakshuki, E. M., & Yoo, W. S. (2020, June). Comparing mobile apps by identifying 'Hot' features. Future Generation Computer Systems, 107, 659 669. https://doi.org/10.1016/j.future.2018.02.008.
[12] Lauff, C., Menold, J., & Wood, K. L. (2019, July). Prototyping Canvas: Design Tool for Planning Purposeful Prototypes. Proceedings of the Design Society: International Conference on Engineering Design, 1(1), 1563–1572. https://doi.org/10.1017/dsi.2019.162
[13] Senarath, U. S. (2021). Waterfall methodology, prototyping and agile development. Tech. Rep., 1-16.

Appendix A: System Architecture Diagram

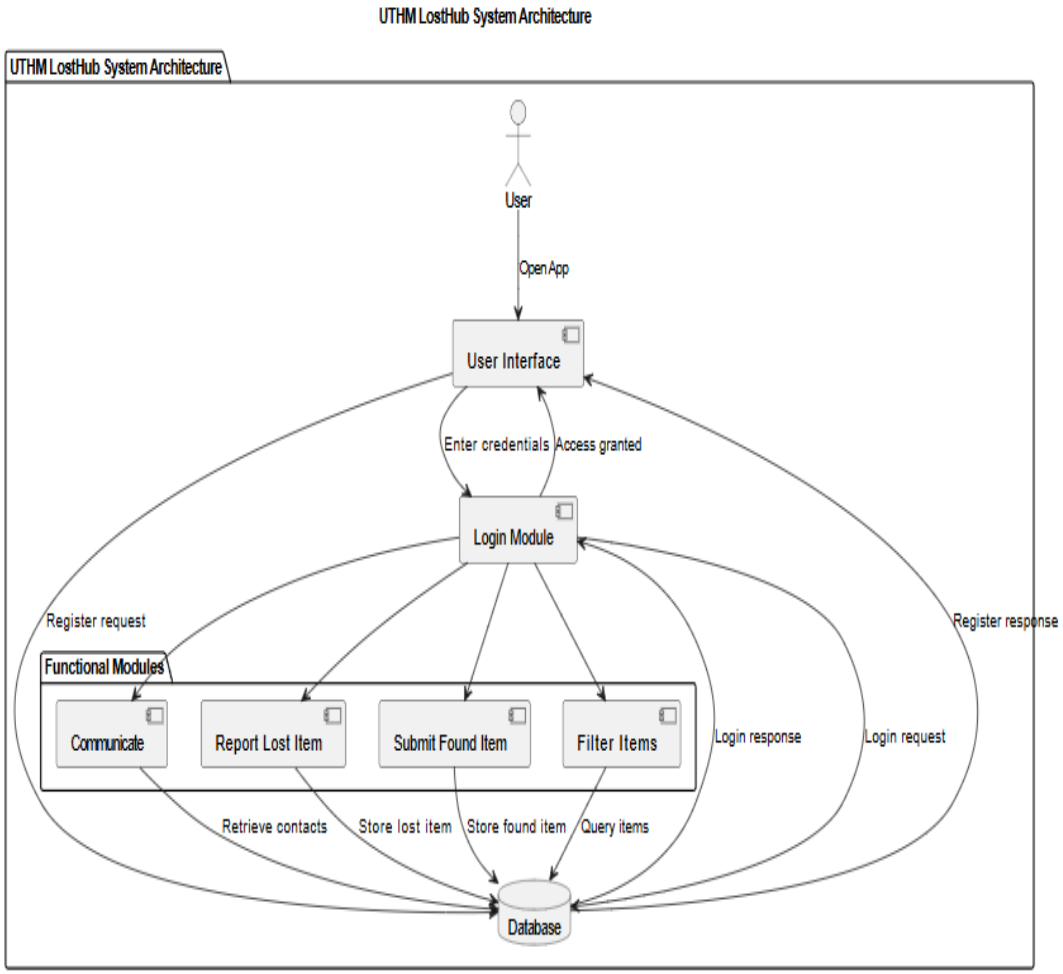


Fig. A.1 *System Architecture Diagram for UTHM Losthub*

Appendix B: Class Diagram

Fig. B.1 *Class Diagram for UTHM Losthub*